Curvature-Balanced Feature Manifold Learning for Long-Tailed Classification

Yanbiao Ma, Licheng Jiao, Fang Liu, Shuyuan Yang, Xu Liu and Lingling Li
Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University
ybmamail@stu.xidian.edu.cn, lchjiao@mail.xidian.edu.cn

Abstract

To address the challenges of long-tailed classification, researchers have proposed several approaches to reduce model bias, most of which assume that classes with few samples are weak classes. However, recent studies have shown that tail classes are not always hard to learn, and model bias has been observed on sample-balanced datasets, suggesting the existence of other factors that affect model bias. In this work, we systematically propose a series of geometric measurements for perceptual manifolds in deep neural networks, and then explore the effect of the geometric characteristics of perceptual manifolds on classification difficulty and how learning shapes the geometric characteristics of perceptual manifolds. An unanticipated finding is that the correlation between the class accuracy and the separation degree of perceptual manifolds gradually decreases during training, while the negative correlation with the curvature gradually increases, implying that curvature imbalance leads to model bias. Therefore, we propose curvature regularization to facilitate the model to learn curvature-balanced and flatter perceptual manifolds. Evaluations on multiple long-tailed and non-long-tailed datasets show the excellent performance and exciting generality of our approach, especially in achieving significant performance improvements based on current state-of-the-art techniques. Our work opens up a geometric analysis perspective on model bias and reminds researchers to pay attention to model bias on non-long-tailed and even sample-balanced datasets. The code and model will be made public.

1. Introduction

The imbalance of sample numbers in the dataset gives rise to the challenge of long-tailed visual recognition. Most previous works assume that head classes are always easier to be learned than tail classes, e.g., class re-balancing [8, 14, 24, 34, 37, 46, 52], information augmentation [23, 31, 35, 38, 39, 44, 56, 64, 67], decoupled training [10, 16, 29, 30, 71, 76], and ensemble learning [20, 36, 57, 58, 61, 72, 77] have been proposed to improve the performance of tail classes. However, recent studies [3, 50] have shown that classification difficulty is not always correlated with the number of samples, e.g., the performance of some tail classes is even higher than that of the head classes. Also, [49] observes differences in model performance across classes on non-long-tailed data, and even on balanced data. Therefore, it is necessary to explore the impact of other inherent characteristics of the data on the classification difficulty, and then improve the overall performance by mitigating the model bias under multiple sample number distribution scenarios.

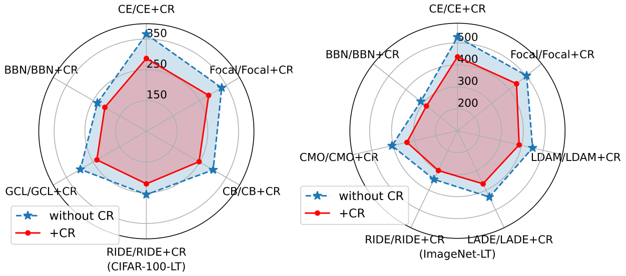

Figure 1. Curvature regularization reduces the model bias present in multiple methods on CIFAR-100-LT and ImageNet-LT. The model bias is measured with the variance of the accuracy of all classes, and it is zero when the accuracy of each class is the same.

Focal loss [37] utilizes the DNN's prediction confidence on instances to evaluate the instance-level difficulty. [50] argues that for long-tailed problems, determining class-level difficulty is more important than determining instance-level difficulty, and therefore defines classification difficulty by evaluating the accuracy of each class in real-time. However, both methods rely on the model output and still can-

This work was supported in part by the Key Scientific Technological Innovation Research Project by Ministry of Education, the State Key Program and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (61836009), the Major Research Plan of the National Natural Science Foundation of China (91438201, 91438103, and 91838303), the National Natural Science Foundation of China (U22B2054, U1701267, 62076192, 62006177, 61902298, 61573267, 61906150, and 62276199), the 111 Project, the Program for Cheung Kong Scholars and Innovative Research Team in University (IRT 15R53), the ST Innovation Project from the Chinese Ministry of Education, the Key Research and Development Program in Shaanxi Province of China(2019ZDLGY03-06), the National Science Basic Research Plan in Shaanxi Province of China(2022JQ-607), the China Postdoctoral fund(2022T150506), the Scientific Research Project of Education Department In Shaanxi Province of China (No.20JY023), the National Natural Science Foundation of China (No. 61977052).

not explain why the model performs well in some classes and poorly in others. Similar to the number of samples, we would like to propose a measure that relies solely on the data itself to model class-level difficulty, which helps to understand how deep neural networks learn from the data. The effective number of samples [14] tries to characterize the diversity of features in each class, but it introduces hyperparameters and would not work in a sample-balanced dataset.

Most data distributions obey the manifold distribution law [33, 54], i.e., samples of each class are distributed near a low-dimensional manifold in the high-dimensional space. The manifold consisting of features in the embedding space is called a perceptual manifold [11]. The classification task is equivalent to distinguishing each perceptual manifold, which has a series of geometric characteristics. We speculate that some geometric characteristics may affect the classification difficulty, and therefore conduct an in-depth study. **The main contributions of our work are: (1)** We systematically propose a series of measurements for the geometric characteristics of point cloud perceptual manifolds in deep neural networks (Sec 3). **(2)** The effect of learning on the separation degree (Sec 4.1) and curvature (Sec 4.2) of perceptual manifolds is explored. We find that the correlation between separation degree and class accuracy decreases with training, while the negative correlation between curvature and class accuracy increases with training (Sec 4.3), implying that existing methods can only mitigate the effect of separation degree among perceptual manifolds on model bias, while ignoring the effect of perceptual manifold complexity on model bias. **(3)** Curvature regularization is proposed to facilitate the model to learn curvature-balanced and flatter feature manifolds, thus improving the overall performance (Sec 5). Our approach effectively reduces the model bias on multiple long-tailed (Fig 1) and non-long-tailed datasets (Fig 8), showing excellent performance (Sec 6).

2. Related Work (Appendix A)

3. The Geometry of Perceptual Manifold

In this section, we systematically propose a series of geometric measures for perceptual manifolds in deep neural networks, and all the pseudocode is in Appendix C.

3.1. Perceptual Manifold

A perceptual manifold is generated when neurons are stimulated by objects with different physical characteristics from the same class. Sampling along the different dimensions of the manifold corresponds to changes in specific physical characteristics. It has been shown [33, 54] that the features extracted by deep neural networks obey the manifold distribution law. That is, features from the same class are distributed near a low-dimensional manifold in the high-dimensional feature space. Given data $X = [x_1, \ldots, x_m]$ from the same class and a deep neural network $Model = \{f(x, \theta_1), g(z, \theta_2)\}$, where $f(x, \theta_1)$ represents a feature sub-network with parameters θ_1 and $g(z, \theta_2)$ represents a classifier with parameters θ_2. Extract the p-dimensional features $Z = [z_1, \ldots, z_m] \in \mathbb{R}^{p\times m}$ of X with the trained model, where $z_i = f(x_i, \theta_1) \in \mathbb{R}^p$. Assuming that the features Z belong to class c, the m features form a p-dimensional point cloud manifold M^c, which is called a perceptual manifold [12].

3.2. The Volume of Perceptual Manifold

We measure the volume of the perceptual manifold M^c by calculating the size of the subspace spanned by the features $z_1, \ldots, z_m$. First, the sample covariance matrix of Z can be estimated as $\Sigma_Z = \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} z_i z_i^T] = \frac{1}{n}ZZ^T \in \mathbb{R}^{p\times p}$. Diagonalize the covariance matrix Σ_Z as UDU^T, where $D = diag(\lambda_1, \ldots, \lambda_p)$ and $U = [u_1, \ldots, u_p] \in \mathbb{R}^{p\times p}$. λ_i and u_i denote the i-th eigenvalue of Σ_Z and its corresponding eigenvector, respectively. Let the singular value of matrix Z be $\sigma_i = \sqrt{\lambda_i}(i = 1, \ldots, p)$. According to the geometric meaning of singular value [1], the volume of the space spanned by vectors $z_1, \ldots, z_m$ is proportional to the product of the singular values of matrix Z, i.e., $Vol(Z) \propto \prod_{i=1}^{p}\sigma_i = \sqrt{\prod_{i=1}^{p}\lambda_i}$. Considering $\lambda_1\lambda_2\cdots\lambda_p = \det(\Sigma_Z)$, the volume of the perceived manifold is therefore denoted as $Vol(Z) \propto \sqrt{\det(\frac{1}{m}ZZ^T)}$.

However, when $\frac{1}{m}ZZ^T$ is a non-full rank matrix, its determinant is 0. For example, the determinant of a planar point set located in three-dimensional space is 0 because its covariance matrix has zero eigenvalues, but obviously the volume of the subspace tensed by the point set in the plane is non-zero. We want to obtain the "area" of the planar point set, which is a generalized volume. We avoid the non-full rank case by adding the unit matrix I to the covariance matrix $\frac{1}{m}ZZ^T$. $I + \frac{1}{m}ZZ^T$ is a positive definite matrix with eigenvalues $\lambda_i + 1(i = 1, \ldots, p)$. The above operation enables us to calculate the volume of a low-dimensional manifold embedded in high-dimensional space. The volume $Vol(Z)$ of the perceptual manifold is proportional to $\sqrt{\det(I + \frac{1}{m}ZZ^T)}$. Considering the numerical stability, we further perform a logarithmic transformation on $\sqrt{\det(I + \frac{1}{m}ZZ^T)}$ and define the volume of the perceptual manifold as

$$Vol(Z) = \frac{1}{2}\log_2 \det(I + \frac{1}{m}(Z - Z_{mean})(Z - Z_{mean})^T),$$

where Z_{mean} is the mean of Z. When $m > 1$, $Vol(Z > 0$. Since $I + \frac{1}{m}(Z - Z_{mean})(Z - Z_{mean})^T$ is a positive definite matrix, its determinant is greater than 0. In the following, the degree of separation between perceptual manifolds will be proposed based on the volume of perceptual manifolds.

3.3. The Separation Degree of Perceptual Manifold

Given the perceptual manifolds M^1 and M^2, they consist of point sets $Z_1 = [z_{1,1}, \ldots, z_{1,m_1}] \in \mathbb{R}^{p \times m_1}$ and $Z_2 = [z_{2,1}, \ldots, z_{2,m_2}] \in \mathbb{R}^{p \times m_2}$, respectively. The volumes of M^1 and M^2 are calculated as $Vol(Z_1)$ and $Vol(Z_2)$. Consider the following case, assuming that M^1 and M^2 have partially overlapped, when $Vol(Z_1) \ll Vol(Z_2)$, it is obvious that the overlapped volume accounts for a larger proportion of the volume of M^1, when the class corresponding to M^1 is more likely to be confused. Therefore, it is necessary to construct an asymmetric measure for the degree of separation between multiple perceptual manifolds, and we expect this measure to accurately reflect the relative magnitude of the degree of separation.

Suppose there are C perceptual manifolds $\{M^i\}_{i=1}^C$, which consist of point sets $\{Z_i = [z_{i,1}, \ldots, z_{i,m_i}] \in \mathbb{R}^{p \times m_i}\}_{i=1}^C$. Let $Z = [Z_1, \ldots, Z_C] \in \mathbb{R}^{p \times \sum_{j=1}^C m_j}$, $Z' = [Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_C] \in \mathbb{R}^{p \times ((\sum_{j=1}^C m_j) - m_i)}$, we define the degree of separation between the perceptual manifold M^i and the rest of the perceptual manifolds as

$$S(M^i) = \frac{Vol(Z) - Vol(Z')}{Vol(Z_i)}.$$

The following analysis is performed for the case when $C = 2$ and $Vol(Z_2) > Vol(Z_1)$. According to our motivation, the measure of the degree of separation between perceptual manifolds should satisfy $S(M^2) > S(M^1)$.

If $S(M^2) > S(M^1)$ holds, then we can get

$$\begin{aligned} & Vol(Z)Vol(Z_1) - Vol(Z_1)^2 > Vol(Z)Vol(Z_2) - Vol(Z_2)^2, \\ \iff & Vol(Z)(Vol(Z_1) - Vol(Z_2)) > Vol(Z_1)^2 - Vol(Z_2)^2, \\ \iff & Vol(Z) < Vol(Z_1) + Vol(Z_2). \end{aligned}$$

We prove that $Vol(Z) < Vol(Z_1) + Vol(Z_2)$ holds when $Vol(Z_2) > Vol(Z_1)$ and the detailed proof is in Appendix B. The above analysis shows that the proposed measure meets our requirements and motivation. The formula for calculating the degree of separation between perceptual manifolds can be further reduced to

$$S(M^i) = \log_\delta \det((I + \frac{Z'Z'^T}{\sum_{j=1, j\neq i}^C m_j})^{-1}(I + \frac{ZZ^T}{\sum_{j=1}^C m_j})),$$

$$\delta = \det(I + \frac{1}{m} Z_i Z_i^T).$$

The detailed derivation is in Appendix B. Next, we validate the proposed measure of the separation degree between perceptual manifolds in a 3D spherical point cloud scene. Specifically, we conducted the experiments in two cases:

(1) Construct two 3D spherical point clouds of radius 1, and then increase the distance between their spherical centers. Since the volumes of the two spherical point clouds are equal, their separation degrees should be symmetric. The variation curves of the separation degrees are plotted in Fig 2, and it can be seen that the experimental results satisfy our theoretical predictions.

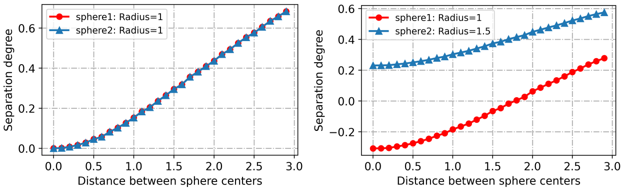

Figure 2. The variation curve between the separation degree of two spherical point clouds and the distance between spherical centers.

(2) Change the distance between the centers of two spherical point clouds of radius 1 and radius 1.5. Observe their separation degrees, the separation degrees of these two spherical point clouds should be asymmetric. Fig 2 shows that their separation degrees increase as the distance between their centers increases. Also, the manifold with a larger radius has a greater separation degree, and this experimental result conforms to our analysis and motivation.

The separation degree between perceptual manifolds may affect the model's bias towards classes. In addition, it can also be used as the regularization term of the loss function or applied in contrast learning to keep the different perceptual manifolds away from each other.

3.4. The Curvature of Perceptual Manifold

Given a point cloud perceptual manifold M, which consists of a p-dimensional point set $\{z_1, \ldots, z_n\}$, our goal is to calculate the Gauss curvature at each point. First, the normal vector at each point on M is estimated by the neighbor points. Denote by z_i^j the j-th neighbor point of z_i and u_i the normal vector at z_i. We solve for the normal vector by minimizing the inner product of $z_i^j - c_i, j = 1, \ldots, k$ and u_i [4], i.e.,

$$\min \textstyle\sum_{j=1}^k ((z_i^j - c_i)^T u_i)^2,$$

where $c_i = \frac{1}{k}\sum_{j=1}^k z_i^j$ and k is the number of neighbor points. Let $y_j = z_i^j - c_i$, then the optimization objective is converted to

$$\begin{aligned} \min \textstyle\sum_{j=1}^k (y_j^T u_i)^2 &= \min \textstyle\sum_{j=1}^k u_i^T y_j y_j^T u_i \\ &= \min(u_i^T (\textstyle\sum_{j=1}^k y_j y_j^T) u_i). \end{aligned}$$

$\sum_{j=1}^k y_j y_j^T$ is the covariance matrix of k neighbors of z_i. Therefore, let $Y = [y_1, \ldots, y_k] \in \mathbb{R}^{p \times k}$ and $\sum_{j=1}^k y_j y_j^T = YY^T$. The optimization objective is further equated to

$$\begin{cases} f(u_i) = u_i^T YY^T u_i, YY^T \in \mathbb{R}^{p \times p}, \\ min(f(u_i)), \\ s.t. u_i^T u_i = 1. \end{cases}$$

Construct the Lagrangian function $L(u_i,\lambda) = f(u_i) - \lambda(u_i^T u_i - 1)$ for the above optimization objective, where λ is a parameter. The first-order partial derivatives of $L(u_i,\lambda)$ with respect to u_i and λ are

$$\begin{aligned}\frac{\partial L(u_i,\lambda)}{\partial u_i} &= \frac{\partial}{\partial u_i} f(u_i) - \lambda \frac{\partial}{\partial u_i}(u_i^T u_i - 1)\\ &= 2(YY^T u_i - \lambda u_i),\\ \frac{\partial L(u_i,\lambda)}{\partial \lambda} &= u_i^T u_i - 1.\end{aligned}$$

Let $\frac{\partial L(u_i,\lambda)}{\partial u_i}$ and $\frac{\partial L(u_i,\lambda)}{\partial \lambda}$ be 0, we can get $YY^T u_i = \lambda u_i, u_i^T u_i = 1$. It is obvious that solving for u_i is equivalent to calculating the eigenvectors of the covariance matrix YY^T, but the eigenvectors are not unique. From $\langle YY^T u_i, u_i \rangle = \langle \lambda u_i, u_i \rangle$ we can get $\lambda = \langle YY^T u_i, u_i \rangle = u_i^T YY^T u_i$, so the optimization problem is equated to $\arg\min_{u_i}(\lambda)$. Performing the eigenvalue decomposition on the matrix YY^T yields p eigenvalues $\lambda_1, \ldots, \lambda_p$ and the corresponding p-dimensional eigenvectors $[\xi_1, \ldots, \xi_p] \in \mathbb{R}^{p\times p}$, where $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$, $\|\xi_i\|_2 = 1, i = 1, \ldots, p$, $\langle \xi_a, \xi_b \rangle = 0 (a \neq b)$. The eigenvector ξ_{m+1} corresponding to the smallest non-zero eigenvalue of the matrix YY^T is taken as the normal vector u_i of M at z_i.

Consider an m-dimensional affine space with center z_i, which is spanned by $\xi_1, \ldots, \xi_m$. This affine space approximates the tangent space at z_i on M. We estimate the curvature of M at z_i by fitting a quadratic hypersurface in the tangent space utilizing the neighbor points of z_i. The k neighbors of z_i are projected into the affine space $z_i + \langle \xi_1, \ldots, \xi_m \rangle$ and denoted as

$$o_j = [(z_i^j - z_i)\cdot \xi_1, \ldots, (z_i^j - z_i)\cdot \xi_m]^T \in \mathbb{R}^m, j = 1, \ldots, k.$$

Denote by $o_j[m]$ the m-th component $(z_i^j - z_i)\cdot \xi_m$ of o_j. We use z_i and k neighbor points to fit a quadratic hypersurface $f(\theta)$ with parameter $\theta \in \mathbb{R}^{m\times m}$. The hypersurface equation is denoted as

$$f(o_j,\theta) = \frac{1}{2}\sum\nolimits_{a,b}\theta_{a,b} o_j\left[a\right] o_j\left[b\right], j \in \left\{1, \ldots, k\right\},$$

further, minimize the squared error

$$E(\theta) = \sum\nolimits_{j=1}^{k}\left(\frac{1}{2}\sum\nolimits_{a,b}\theta_{a,b} o_j\left[a\right] o_j\left[b\right] - (z_i^j - z_i)\cdot u_i\right)^2.$$

Let $\frac{\partial E(\theta)}{\partial \theta_{a,b}} = 0, a, b \in \{1, \ldots, m\}$ yield a nonlinear system of equations, but it needs to be solved iteratively. Here, we propose an ingenious method to fit the hypersurface and **give the analytic solution of the parameter** θ directly. Expand the parameter θ of the hypersurface into the column vector

$$\theta = [\theta_{1,1}, \ldots, \theta_{1,m}, \theta_{2,1}, \ldots, \theta_{m,m}]^T \in \mathbb{R}^{m^2}.$$

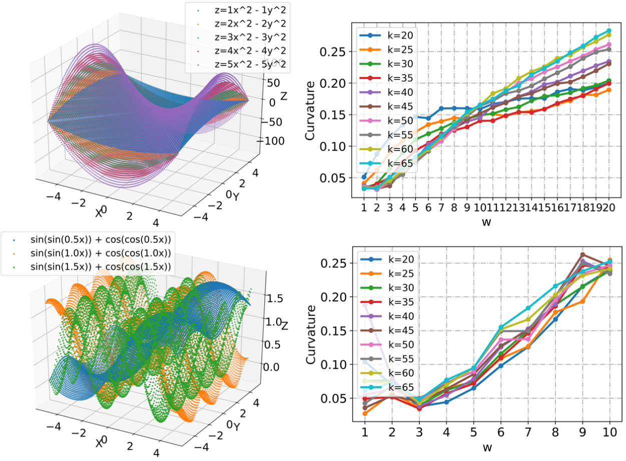

Figure 3. The surface equations in the first and second rows are $Z = w(X^2 - Y^2)$ and $Z = \sin(\sin(0.5wX)) + \cos(\cos(0.5wX))$, respectively. We increase the curvature of the surface by increasing w and calculate the complexity of the two-dimensional point cloud surface. Also, we investigate the effect of the number of neighbors k on the complexity of the manifold.

Organize the k neighbor points $\{o_j\}_{j=1}^k$ of z_i according to the following form:

$$O(z_i) = \begin{bmatrix} o_1\left[1\right] o_1\left[1\right] & o_1\left[1\right] o_1\left[2\right] & \cdots & o_1\left[m\right] o_1\left[m\right] \\ o_2\left[1\right] o_2\left[1\right] & o_2\left[1\right] o_2\left[2\right] & \cdots & o_2\left[m\right] o_2\left[m\right] \\ \vdots & \vdots & \ddots & \vdots \\ o_k\left[1\right] o_k\left[1\right] & o_k\left[1\right] o_k\left[2\right] & \cdots & o_k\left[m\right] o_k\left[m\right] \end{bmatrix} \in \mathbb{R}^{k\times m^2}.$$

The target value is

$$T = \left[(z_i^1 - z_i)\cdot u_i, (z_i^2 - z_i)\cdot u_i, \ldots, (z_i^k - z_i)\cdot u_i\right]^T \in \mathbb{R}^k.$$

We minimize the squared error

$$E(\theta) = \frac{1}{2} tr\left[\left(O(z_i)\theta - T\right)^T \left(O(z_i)\theta - T\right)\right],$$

and find the partial derivative of $E(\theta)$ for θ:

$$\begin{aligned}\frac{\partial E(\theta)}{\partial \theta} &= \frac{1}{2}\left(\frac{\partial tr(\theta^T O(z_i)^T O(z_i)\theta)}{\partial \theta} - \frac{\partial tr(\theta^T O(z_i)^T T)}{\partial \theta}\right)\\ &= O(z_i)^T O(z_i)\theta - O(z_i)^T T.\end{aligned}$$

Let $\frac{\partial E(\theta)}{\partial \theta} = 0$, we can get

$$\theta = (O(z_i)^T O(z_i))^{-1} O(z_i)^T T.$$

Thus, the Gauss curvature of the perceptual manifold M at z_i can be calculated as

$$G(z_i) = det(\theta) = det((O(z_i)^T O(z_i))^{-1} O(z_i)^T T).$$

Up to this point, we provide an approximate solution of the Gauss curvature at any point on the point cloud perceptual manifold M. [5] shows that on a high-dimensional dataset, almost all samples lie on convex locations, and thus the complexity of the perceptual manifold is defined as the average $\frac{1}{n}\sum_{i=1}^{n}G(z_i)$ of the Gauss curvatures at all points on M. Our approach does not require iterative optimization and can be quickly deployed in a deep neural network to calculate the Gauss curvature of the perceptual manifold. Taking the two-dimensional surface in Fig 3 as an example, the surface complexity increases as the surface curvature is artificially increased. This indicates that our proposed complexity measure of perceptual manifold can accurately reflect the changing trend of the curvature degree of the manifold. In addition, Fig 3 shows that the selection of the number of neighboring points hardly affects the monotonicity of the complexity of the perceptual manifold. In our work, we select the number of neighboring points to be 40.

4. Learning How to Shape Perceptual Manifold

The perceptual manifolds in feature space are further decoded by the classification network into predicted probabilities for classification. Intuitively, we speculate that a perceptual manifold is easier to be decoded by the classification network when it is farther away from other perceptual manifolds and flatter. We provide more geometric views on classification and object detection in Appendix I. A model is usually considered to be biased when its performance on classes is inconsistent. In the following, we investigate the effect of the geometry of the perceptual manifold on the model bias and summarize three experimental discoveries.

4.1. Learning Facilitates The Separation

Learning typically leads to greater inter-class distance, which equates to greater separation between perceptual manifolds. We trained VGG-16 [48] and ResNet-18 [22] on F-MNIST [62] and CIFAR-10 [32] to explore the effect of the learning process on the separation degree between perceptual manifolds and observed the following phenomenon.

As shown in Fig 4, each perceptual manifold is gradually separated from the other manifolds during training. It is noteworthy that the separation is faster in the early stage of training, and the increment of separation degree gradually decreases in the later stage. Separation curves of perceptual manifolds for more classes are presented in Appendix D.

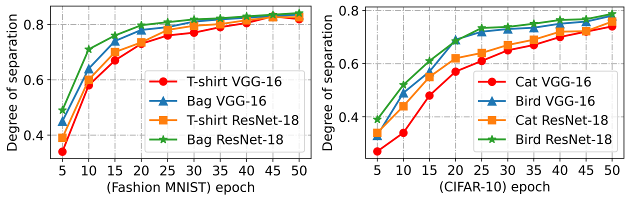

Figure 4. The variation curves between the separation degree of perceptual manifolds and training epochs on both datasets.

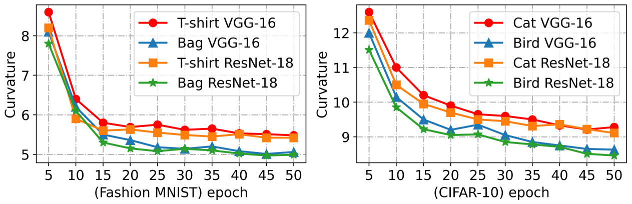

Figure 5. The variation curves between the complexity of perceptual manifolds and training epochs on both datasets.

4.2. Learning Reduces The Curvature

Experiments are conducted with VGG-16 and ResNet-18 trained on F-MNIST and CIFAR-10, and we find that the perceptual manifold gradually flattens out during training. As shown in Fig 5, the curvature of the perceptual manifold decreases faster in the early stage of training, and it gradually becomes flat with further training. The curvature change curves of perceptual manifolds for more classes are shown in Appendix E.

4.3. Curvature Imbalance and Model Bias

Since learning separates perceptual manifolds from each other and also makes perceptual manifolds flatter, it is reasonable to speculate that the separation degree and curvature of perceptual manifolds correlate with class-wise classification difficulty. Experiments are conducted with VGG-16 and ResNet-18 trained on F-MNIST and CIFAR-10.

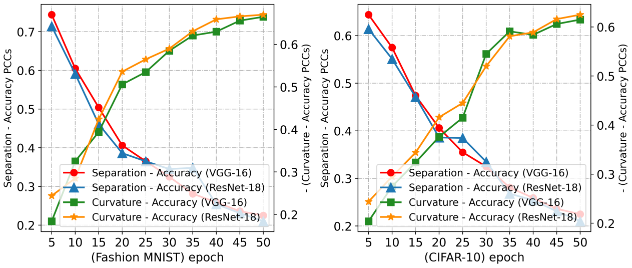

Figure 6. The Pearson correlation coefficients (PCCs) between the accuracy of all classes and the separation degree and complexity of the corresponding perceptual manifolds, respectively.

Each class corresponds to a perceptual manifold. As shown in Fig 6, we observe that the negative correlation between the separation degree of the perceptual manifolds and the accuracy of the corresponding class decreases with training, while the correlation between the curvature and the accuracy increases. This implies that existing methods can only mitigate the effect of the separation degree between perceptual manifolds on the model bias, while ignoring the effect of perceptual manifold complexity on the model bias.

5. Curvature-Balanced Feature Learning

The above study shows that it is necessary to focus on the model bias caused by the curvature imbalance among perceptual manifolds. In this section, we propose curvature regularization, which can reduce the model bias and further improve the performance of existing methods.

5.1. Design Principles of The Proposed Approach

The proposed curvature regularization needs to satisfy the following three principles to learn curvature-balanced and flat perceptual manifolds.

(1) The greater the curvature of a perceptual manifold, the stronger the penalty for it. Our experiments show that learning reduces the curvature, so it is reasonable to assume that flatter perceptual manifolds are easier to decode. **(2)** When the curvature is balanced, the penalty strength is the same for each perceptual manifold. **(3)** The sum of the curvatures of all perceptual manifolds tends to decrease.

5.2. Curvature Regularization (CR)

Given a C classification task, the p-dimensional feature embeddings of images from each class are represented as $Z_i = \left[z_i^1, \ldots, z_i^{m_i}\right], i = 1, \ldots, C$. The mean Gaussian curvature $G_i, i = 1, \ldots, C$ of the corresponding perceptual manifold is calculated with the feature embeddings of each class (Appendix C.Algorithm 5). First, take the inverse of the curvature G_i and perform the maximum normalization on it. Then the negative logarithmic transformation is executed on the normalized curvature, and the curvature penalty term of the perceptual manifold M^i is $-\log(\frac{G_i^{-1}}{\max\{G_1^{-1},\ldots,G_C^{-1}\}})$. Further, the overall curvature regularization term is denoted as

$$L_{Curvature} = \textstyle\sum_{i=1}^{C} -\log(\frac{G_i^{-1}}{\max\{G_1^{-1},\ldots,G_C^{-1}\}}).$$

The detailed derivation is shown in Appendix F. In the following, we verify whether $L_{Curvature}$ satisfies the three principles one by one.

(1) When the curvature G_i of the perceptual manifold is larger, G_i^{-1} is smaller. Since $-\log(\cdot)$ is monotonically decreasing, $-\log(\frac{G_i^{-1}}{\max\{G_1^{-1},\ldots,G_C^{-1}\}})$ increases with G_i increases. $L_{Curvature}$ is consistent with Principle 1.

(2) When $G_1 = \cdots = G_C$, $\max\{G_1^{-1}, \ldots, G_C^{-1}\} = G_1^{-1} = \cdots = G_C^{-1}$, so $-\log(\frac{G_i^{-1}}{\max\{G_1^{-1},\ldots,G_C^{-1}\}}) = 0, i = 1, \ldots, C$. $L_{Curvature}$ follows Principle 2.

(3) The curvature penalty term of the perceptual manifold M^i is 0 when $G_i = \min\{G_1, \ldots, G_C\}$. Since the greater the curvature, the greater the penalty, our method aims to bring the curvature of all perceptual manifolds down to $\min\{G_1, \ldots, G_C\}$. Obviously, $\sum_{i=1}^{C} G_i \geq C \cdot \min\{G_1, \ldots, G_C\}$, so our approach promotes curvature balance while also making all perceptual manifolds flatter, which satisfies Principle 3.

The curvature regularization can be combined with any loss function. Since the correlation between curvature and accuracy increases with training, we balance the curvature regularization with other losses using a logarithmic function with a hyperparameter τ, and the overall loss is denoted as

$$L = L_{original} + \frac{\log_\tau epoch}{(\frac{L_{Curvature}}{L_{original}}).detach()} \times L_{Curvature},\ \tau > 1.$$

The term $(\frac{L_{Curvature}}{L_{original}}).detach()$ aims to make the curvature regularization loss of the same magnitude as the original loss. We investigate reasonable values of τ in experiments (Sec 6.2). The design principle of curvature regularization is compatible with the learning objective of the model, and our experiments show that the effect of curvature imbalance on model bias has been neglected in the past. Thus curvature regularization is not in conflict with $L_{original}$, as evidenced by our outstanding performance on multiple datasets.

5.3. Dynamic Curvature Regularization (DCR)

The curvature of perceptual manifolds varies with the model parameters during training, so it is necessary to update the curvature of each perceptual manifold in real-time. However, there is a challenge: only one batch of features is available at each iteration, and it is not possible to obtain all the features to calculate the curvature of the perceptual manifolds. If the features of all images from the training set are extracted using the current network at each iteration, it will greatly increase the time cost of training.

Algorithm 1 End-to-end training with DCR

```
Require: Training set $D = \{(x_i, y_i)\}_{i=1}^{M}$. A CNN $\{f(x,\theta_1), g(z,\theta_2)\}$, where $f(\cdot)$ and $g(\cdot)$ denote the feature sub-network and classifier, respectively. The training epoch is $N$.
Initialize the storage pool Q
for $epoch = 1$ to $N$ do
    for $iteration = 0$ to $\frac{M}{batch\ size}$ do
        Sample a mini-batch $\{(x_i, y_i)\}_{i=1}^{batch\ size}$ from $D$.
        Calculate feature embeddings $z_i = f(x_i, \theta_1), i = 1, \ldots, batch\ size$.
        Store $z_i$ and label $y_i$ into $Q$.
        if $epoch < n$ then
            if $epoch > 1$ then
                Dequeue the oldest batch of features from $Q$.
            end if
            Calculate loss $L_{original}$.
        else
            Dequeue the oldest batch of features from $Q$.
            Calculate the curvature of each perceptual manifold.
            Calculate loss:
            $L = L_{original} + \frac{\log_\tau epoch}{(\frac{L_{Curvature}}{L_{original}}).detach()} \times L_{Curvature}$.
        end if
        Perform back propagation: $L.backward()$.
        $optimizer.step()$.
    end for
end for
```

Inspired by [3, 40], we design a first-in-first-out storage pool to store the latest historical features of all images. The slow drift phenomenon of features found by [59] ensures the reliability of using historical features to approximate the current features. We show the training process in Algorithm 1. Specifically, the features of all batches are stored in the storage pool at the first epoch. To ensure that the drift of the features is small enough, it is necessary to train another n epochs to update the historical features. Experiments of [3] on large-scale datasets show that n taken as 5 is sufficient, so n is set to 5 in this work. When $epoch > n$, the oldest batch of features in the storage pool is replaced with new features at each iteration, and the curvature of each perceptual manifold is calculated using all features in the storage pool. The curvature regularization term is updated based on the latest curvature. **It should be noted** that for decoupled training, CR is applied in the feature learning stage. Our method is employed in training only and does not affect the inference speed of the model.

6. Experiments

6.1. Datasets and Implementation Details

We comprehensively evaluate the effectiveness and generality of curvature regularization on both long-tailed and non-long-tailed datasets. The experiment is divided into two parts, the first part tests curvature regularization on four long-tailed datasets, namely CIFAR-10-LT, CIFAR-100-LT [14], ImageNet-LT [14, 47], and iNaturalist2018 [55]. The second part validates the curvature regularization on two non-long tail datasets, namely CIFAR-100 [32] and ImageNet [47]. For a fair comparison, the training and test images of all datasets are officially split, and the Top-1 accuracy on the test set is utilized as a performance metric. In addition, we train models on CIFAR-100, CIFAR-10/100-LT with a single NVIDIA 2080Ti GPU and ImageNet, ImageNet-LT, and iNaturalist2018 with eight NVIDIA 2080Ti GPUs. Please refer to Appendix G for a detailed description of the dataset and experimental setup.

6.2. Effect of τ

When $\tau = epoch$, $\log_\tau epoch = 1$, so the selection of τ is related to the number of epochs. When the correlation between curvature and accuracy exceeds the correlation between the separation degree and accuracy, we expect $\log_\tau epoch > 1$, which means that the curvature regularization loss is greater than the original loss. Following the [45] setting, all models are trained for 200 epochs, so τ is less than 200. To search for the proper value of τ, experiments are conducted for CE + CR with a range of τ, and the results are shown in Fig 7. Large-scale datasets require more training epochs to keep the perceptual manifolds away from each other, while small-scale datasets can achieve this faster, so we set $\tau = 100$ on CIFAR-10/100-LT and CIFAR-100, and $\tau = 120$ on ImageNet, ImageNet-LT, and iNaturalist2018.

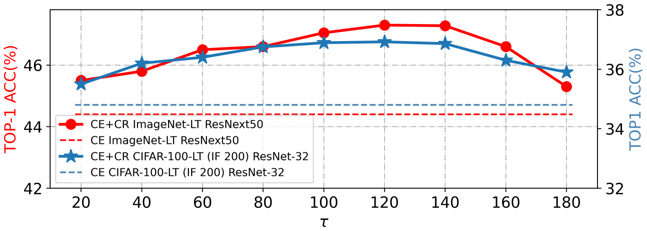

Figure 7. The effect of τ on accuracy for both datasets.

Table 1. Comparison on CIFAR-10-LT and CIFAR-100-LT. The accuracy (%) of Top-1 is reported. The best and second-best results are shown in **underlined bold** and **bold**, respectively.

Dataset	CIFAR-10-LT				CIFAR-100-LT			
Backbone Net	ResNet-32							
imbalance factor	200	100	50	10	200	100	50	10
MiSLAS [76]	77.3	82.1	**85.7**	**90.0**	42.3	47.0	52.3	**63.2**
LDAM-DRW [8]	-	77.0	81.0	88.2	-	42.0	46.6	58.7
Cross Entropy	65.6	70.3	74.8	86.3	34.8	38.2	43.8	55.7
+ CR	67.9	72.6	76.2	89.5	36.9	40.5	45.1	57.4
Focal Loss [37]	65.2	70.3	76.7	86.6	35.6	38.4	44.3	55.7
+ CR	67.3	71.8	79.1	88.4	37.5	40.2	45.2	58.3
CB Loss [14]	68.8	74.5	79.2	87.4	36.2	39.6	45.3	57.9
+ CR	70.3	75.8	79.8	89.1	38.5	40.7	46.8	59.2
BBN [77]	-	79.8	82.1	88.3	-	42.5	47.0	59.1
+ CR [34]	-	81.2	83.5	89.4	-	43.7	48.1	60.0
De-c-TDE [53]	-	80.6	83.6	88.5	-	44.1	50.3	59.6
+ CR	-	81.8	84.5	**89.9**	-	45.7	51.4	60.3
RIDE (4*) [58]	-	-	-	-	-	48.7	**59.0**	58.4
+ CR	-	-	-	-	-	49.8	**59.8**	59.5
RIDE + CMO [45]	-	-	-	-	-	**50.0**	53.0	60.2
+ CR	-	-	-	-	-	**50.7**	54.3	**61.4**
GCL [34]	**79.0**	**82.7**	85.5	-	**44.9**	48.7	53.6	-
+ CR	**79.9**	**83.5**	**86.8**	-	**45.6**	49.8	55.1	-

6.3. Experiments on Long-Tailed Datasets

6.3.1 Evaluation on CIFAR-10/100-LT

Table 1 summarizes the improvements of CR for several state-of-the-art methods on long-tailed CIFAR-10 and CIFAR-100, and we observe that CR significantly improves all methods. For example, in the setting of IF 200, CR results in performance gains of 2.3%, 2.1%, and 1.5% for CE, Focal loss [37], and CB loss [14], respectively. When CR is applied to feature training, the performance of BBN [77] is improved by more than 1% on each dataset, which again validates that curvature imbalance negatively affects the learning of classifiers. When CR is applied to several state-of-the-art methods (e.g., RIDE + CMO [45] (2022) and GCL [34] (2022)), CR achieves higher classification accuracy with all IF settings.

Table 2. Top-1 accuracy (%) of ResNext-50 [63] on ImageNet-LT and Top-1 accuracy (%) of ResNet-50 [22] on iNaturalist2018 for classification. The best and the second-best results are shown in **underline bold** and **bold**, respectively.

Methods	ImageNet-LT				iNaturalist 2018			
	ResNext-50				ResNet-50			
	H	M	T	Overall	H	M	T	Overall
OFA [10]	47.3	31.6	14.7	35.2	-	-	-	65.9
DisAlign [71]	59.9	49.9	31.8	52.9	68.0	71.3	69.4	70.2
MiSLAS [76]	65.3	50.6	33.0	53.4	**73.2**	72.4	70.4	71.6
DiVE [23]	64.0	50.4	31.4	53.1	70.6	70.0	67.5	69.1
PaCo [13]	63.2	51.6	39.2	54.4	69.5	72.3	73.1	72.3
GCL [34]	-	-	-	54.9	-	-	-	72.0
CE	65.9	37.5	7.70	44.4	67.2	63.0	56.2	61.7
+ CR	65.1	40.7	19.5	47.3	67.3	62.6	61.7	63.4
Focal Loss [37]	67.0	41.0	13.1	47.2	-	-	-	61.1
+ CR	67.3	43.2	22.5	49.6	69.4	61.7	57.2	62.3
BBN [77]	43.3	45.9	**43.7**	44.7	49.4	70.8	65.3	66.3
+ CR	45.2	46.8	**44.5**	46.2	50.6	71.5	66.8	67.6
LDAM [8]	60.0	49.2	31.9	51.1	-	-	-	64.6
+ CR	60.8	50.3	33.6	52.4	69.3	66.7	61.9	65.7
LADE [24]	62.3	49.3	31.2	51.9	-	-	-	69.7
+ CR	62.5	50.1	33.7	53.0	72.5	70.4	65.7	70.6
MBJ [40]	61.6	48.4	39.0	52.1	-	-	-	70.0
+ CR	62.8	49.2	40.4	53.4	**73.1**	70.3	66.0	70.8
RIDE (4*) [58]	**67.8**	53.4	36.2	56.6	70.9	72.4	73.1	72.6
+ CR	**68.5**	54.2	38.8	**57.8**	71.0	**73.8**	**74.3**	**73.5**
RIDE + CMO [45]	66.4	**54.9**	35.8	56.2	70.7	72.6	73.4	72.8
+ CR	67.3	**54.6**	38.4	**57.4**	71.6	**73.7**	**74.9**	**73.8**

6.3.2 Evaluation on ImageNet-LT and iNaturalist2018

The results on ImageNet-LT and iNaturalist2018 are shown in Table 2. We not only report the overall performance of CR, but also additionally add the performance on three subsets of Head (more than 100 images), Middle (20-100 images), and Tail (less than 20 images). From Table 2, we observe the following three conclusions: first, CR results in significant overall performance improvements for all methods, including 2.9% and 2.4% improvements on ImageNet-LT for CE and Focal loss, respectively. Second, when CR is combined with feature training, the overall performance of BBN [77] is improved by 1.5% and 1.3% on the two datasets, respectively, indicating that curvature-balanced feature learning facilitates classifier learning. Third, our approach still boosts model performance when combined with advanced techniques (RIDE [58] (2021), RIDE + CMO [45] (2022)), suggesting that curvature-balanced feature learning has not yet been considered by other methods.

6.4. Experiments on Non-Long-Tailed Datasets

Curvature imbalance may still exist on sample-balanced datasets, so we evaluate CR on non-long-tailed datasets. Table 3 summarizes the improvements of CR on CIFAR-100 and ImageNet for various backbone networks, and we observe that CR results in approximately 1% performance improvement for all backbone networks. In particular, the accuracy of CE + CR exceeds CE by 1.5% on CIFAR-100 when using ResNet-18 [22] as the backbone network. The experimental results show that our proposed curvature regularization is applicable to non-long-tailed datasets and compatible with existing backbone networks and methods.

Table 3. Comparison on ImageNet and CIFAR-100.

	ImageNet			CIFAR-100		
Methods	CE	CE + CR	Δ	CE	CE + CR	Δ
VGG16 [48]	71.6	72.7	+1.1	71.9	73.2	+1.3
BN-Inception [51]	73.5	74.3	+0.8	74.1	75.0	+0.9
ResNet-18 [22]	70.1	71.3	+1.2	75.6	77.1	+1.5
ResNet-34 [22]	73.5	74.6	+1.1	76.8	78.0	+1.2
ResNet-50 [22]	76.0	76.8	+0.8	77.4	78.3	+0.9
DenseNet-201 [28]	77.2	78.0	+0.8	78.5	79.7	+1.2
SE-ResNet-50 [25]	77.6	78.3	+0.7	78.6	79.5	+0.9
ResNeXt-101 [63]	78.8	79.7	+0.9	77.8	78.9	+1.1

6.5. Curvature Regularization Reduces Model Bias

Here we explore how curvature regularization improves the model performance. Measuring the model bias with the variance of the accuracy of all classes [50], Fig 1 and Fig 8 show that curvature regularization reduces the bias of the models trained on CIFAR-100-LT, Image-Net-LT, CIFAR-100, and ImageNet. By combining Tables 1 and 2, it can be found that curvature regularization reduces the model bias mainly by improving the performance of the tail class and does not compromise the performance of the head class, thus improving the overall performance. In addition, In Appendix H we answer the following two questions: (1) Is the curvature more balanced after training with CR? (2) Did the correlation between curvature imbalance and class accuracy decrease after training with CR?

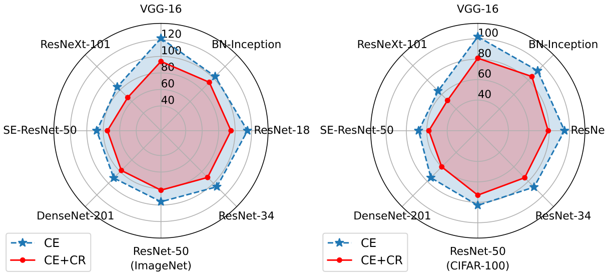

Figure 8. Curvature regularization reduces the bias of multiple backbone networks trained on ImageNet and CIFAR-100.

7. Conclusion

This work mines and explains the impact of data on the model bias from a geometric perspective, introducing the imbalance problem to non-long-tailed data and providing a geometric analysis perspective to drive toward fairer AI.

References

[1] Charu C Aggarwal, Lagerstrom-Fife Aggarwal, and Lagerstrom-Fife. *Linear algebra and optimization for machine learning*, volume 156. Springer, 2020. 2

[2] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6897–6907, 2022. 13

[3] Anonymous. Delving into semantic scale imbalance. In *Submitted to The Eleventh International Conference on Learning Representations*, 2023. under review. 1, 7, 13, 14

[4] Yasuhiko Asao and Yuichi Ike. Curvature of point clouds through principal component analysis. *arXiv preprint arXiv:2106.09972*, 2021. 3

[5] Randall Balestriero, Jerome Pesenti, and Yann LeCun. Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485*, 2021. 4

[6] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 14

[7] Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 112–121, 2021. 14

[8] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019. 1, 7, 8, 19

[9] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 13

[10] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *European Conference on Computer Vision*, pages 694–710. Springer, 2020. 1, 8, 13

[11] SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Linear readout of object manifolds. *Physical Review E*, 93(6):060301, 2016. 2

[12] Uri Cohen, SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature communications*, 11(1):1–13, 2020. 2, 16

[13] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 715–724, 2021. 8, 14

[14] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 1, 2, 7, 13, 19

[15] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4109–4118, 2018. 13

[16] Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. Learning of visual relations: The devil is in the tails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15404–15413, 2021. 1

[17] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1851–1860, 2017. 14

[18] Charles Elkan. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001. 13

[19] Andrew Estabrooks, Taeho Jo, and Nathalie Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational intelligence*, 20(1):18–36, 2004. 13

[20] Hao Guo and Song Wang. Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15089–15098, 2021. 1

[21] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005. 13

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 8, 19

[23] Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. Distilling virtual examples for long-tailed recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 235–244, 2021. 1, 8

[24] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on*

computer vision and pattern recognition, pages 6626–6636, 2021. 1, 8

[25] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 8

[26] Xinting Hu, Yi Jiang, Kaihua Tang, Jingyuan Chen, Chunyan Miao, and Hanwang Zhang. Learning to segment the tail. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14045–14054, 2020. 13

[27] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016. 14

[28] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 8

[29] Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2020. 1, 14

[30] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019. 1, 13

[31] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13896–13905, 2020. 1, 13

[32] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5, 7

[33] Na Lei, Dongsheng An, Yang Guo, Kehua Su, Shixia Liu, Zhongxuan Luo, Shing-Tung Yau, and Xianfeng Gu. A geometric understanding of deep learning. *Engineering*, 6(3):361–374, 2020. 2, 16

[34] Mengke Li, Yiu-ming Cheung, and Yang Lu. Long-tailed visual recognition via gaussian clouded logit adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6929–6938, 2022. 1, 7, 8

[35] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5212–5221, 2021. 1, 14

[36] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10991–11000, 2020. 1

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1, 7, 8, 13

[38] Bo Liu, Haoxiang Li, Hao Kang, Gang Hua, and Nuno Vasconcelos. Gistnet: a geometric structure transfer network for long-tailed recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8209–8218, 2021. 1, 13

[39] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2970–2979, 2020. 1, 13

[40] Jialun Liu, Jingwei Zhang, Wenhui Li, Chi Zhang, Yifan Sun, et al. Memory-based jitter: Improving visual recognition on long-tailed data with diversity in memory. *arXiv preprint arXiv:2008.09809*, 2020. 7, 8

[41] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018. 13

[42] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. 13

[43] Wanli Ouyang, Xiaogang Wang, Cong Zhang, and Xiaokang Yang. Factors in finetuning deep model for object detection with long-tail distribution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 864–873, 2016. 14

[44] Sarah Parisot, Pedro M Esperança, Steven McDonagh, Tamas J Madarasz, Yongxin Yang, and Zhenguo Li. Long-tail recognition via compositional knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6939–6948, 2022. 1

[45] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6887–6896, 2022. 7, 8, 13

[46] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020. 1, 13

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 7, 19

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5, 8

[49] Saptarshi Sinha, Hiroki Ohashi, and Katsuyuki Nakamura. Class-wise difficulty-balanced loss for solving class-imbalance. In *Proceedings of the Asian conference on computer vision*, 2020. 1, 13

[50] Saptarshi Sinha, Hiroki Ohashi, and Katsuyuki Nakamura. Class-difficulty based methods for long-tailed visual recognition. *International Journal of Computer Vision*, 130(10):2517–2531, 2022. 1, 8, 13, 14

[51] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 8

[52] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020. 1, 13

[53] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in Neural Information Processing Systems*, 33:1513–1524, 2020. 7

[54] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000. 2

[55] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 7, 19

[56] Jianfeng Wang, Thomas Lukasiewicz, Xiaolin Hu, Jianfei Cai, and Zhenghua Xu. Rsg: A simple but effective module for learning imbalanced datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3784–3793, 2021. 1

[57] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *European conference on computer vision*, pages 728–744. Springer, 2020. 1, 13, 14

[58] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020. 1, 7, 8, 13, 14

[59] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6388–6397, 2020. 7

[60] Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5017–5026, 2019. 13

[61] Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *European Conference on Computer Vision*, pages 247–263. Springer, 2020. 1

[62] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[63] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 8, 19

[64] Zhengzhuo Xu, Zenghao Chai, and Chun Yuan. Towards calibrated model for long-tailed visual recognition from prior perspective. *Advances in Neural Information Processing Systems*, 34:7139–7152, 2021. 1

[65] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning.

Advances in neural information processing systems, 33:19290–19301, 2020. 13

[66] Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *arXiv preprint arXiv:2001.01385*, 2020. 13

[67] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5704–5713, 2019. 1, 13

[68] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 19

[69] Yuhang Zang, Chen Huang, and Chen Change Loy. Fasa: Feature augmentation and sampling adaptation for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3457–3466, 2021. 13

[70] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 19

[71] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021. 1, 8

[72] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. *arXiv preprint arXiv:2107.09249*, 2021. 1, 14

[73] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021. 13

[74] Zizhao Zhang and Tomas Pfister. Learning fast sample re-weighting without reward data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 725–734, 2021. 13

[75] Peilin Zhao, Yifan Zhang, Min Wu, Steven CH Hoi, Mingkui Tan, and Junzhou Huang. Adaptive cost-sensitive online classification. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):214–228, 2018. 13

[76] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16489–16498, 2021. 1, 7, 8, 13

[77] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020. 1, 7, 8, 13, 14

[78] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1):63–77, 2005. 13